

Chapter Twelve: Contents

(RS-m (Blue) – 10 December 2002 – LA-UR 01-5716 – Portland Study Reports)

1. CONFIGURATION FILES	1
1.1 ALLSTR_ROUTER_RS12-FB.CFG	1
2. SCRIPTS	2
2.1 COMBINESUMMARYFILES.PL	2
2.2 CLEANPLAN.....	4
2.3 CREATEPARTIALHHFILE.....	5
2.4 TRAVTOHH.PL	7
2.5 DISTRIB.SH	8
2.6 MKINDEX.SH	9
2.7 MVPLANS.SH	10
2.8 VALIDIDX.SH.....	11

Chapter Twelve—RS-m (Blue)

NOTE: Long code lines that do not fit completely on one line of this document are shown in italics and continued on to the next line.

1. CONFIGURATION FILES

1.1 allstr_router_RS12-FB.cfg

```
TRANSIMS_ROOT           /n/father/transims/CaseStudy3/scenarios/allstr
CONFIG_DEFAULT_FILE     $TRANSIMS_ROOT/allstr.cfg

ACTIVITY_FILE           $TRANSIMS_ROOT/activity/AS7
PLAN_FILE               $TRANSIMS_ROOT/plans/RS12R.FB/plans.RS12.FB
ROUTER_HOUSEHOLD_FILE  $TRANSIMS_ROOT/router/RS12R/hh.RS12.FB
ROUTER_COMPLETED_HOUSEHOLD_FI $TRANSIMS_ROOT/router/RS12R/hh.done.RS12.FB
ROUTER_PROBLEM_FILE    $TRANSIMS_ROOT/router/RS12R/problems.RS12.FB
ROUTER_LINK_DELAY_FILE $TRANSIMS_ROOT/router/RS12R/summary.dens.RS12.FB.fixed.tim
```

2. SCRIPTS

2.1 CombineSummaryFiles.pl

```
#!/usr/local/gnu/bin/perl -w

sub readLinkLengths;

if ($#ARGV != 4)
{
    print $#ARGV, "\n";
    die "Usage: $0 <config file> <summary file> <noflow file> <output file> <speed>\n";
}

$configfile = $ARGV[0];
$summaryfile = $ARGV[1];
$noflowfile = $ARGV[2];
$outputfile = $ARGV[3];
$speed = $ARGV[4];
print "speed: $speed\n";

readLinkLengths($configfile);

`cp $summaryfile $outputfile`;

open $nff, $noflowfile;
open $of, ">>$outputfile";
while (<$nff>)
{
    if ($. > 2)
    {
        ($count, $lane, $link, $node, $sum, $sumsquares, $time, $turn,
         $vcount, $vsum, $vsumsquares) = (split '\t', $_);
        $count = 1;
        $sum = $delay{$link};
        print $of "$count\t$lane\t$link\t$node\t$sum\t$sumsquares\t$time\t$turn\t",
            "$vcount\t$vsum\t$vsumsquares";
    }
}
close $nff;

sub readLinkLengths
{
    my $cf = $_[0];
```

```
print "Evaluating config file: $cf\n";
my $config = `~/home/transims/CaseStudy3/bin/SetEnv perl $cf | tail +2 | grep -v "Success exit"`;
eval $config;
my $linktbl = $CommonConfig{"NET_DIRECTORY"} . $CommonConfig{"NET_LINK_TABLE"};
print "Reading link table: $linktbl\n";
my $lf;
open $lf, $linktbl;
while (<$lf>)
{
    if ($. != 1)
    {
        my $id = (split '\t', $_)[0];
        my $length = (split '\t', $_)[11];
        $delay{$id} = int $length/$speed;
        print "$id $length $delay{$id}\n";
#         ($node, $num10, $num26) = split ' ', $_;
    }
}
close $lf;
}
```

2.2 CleanPlan

```
#!/bin/csh

# Usage: CleanPlan <plan1> <plan2> <dest plan>

# Clean up plans from a restarted router.  The problem is that there
# may be plans in a plan file from an aborted router that are not in
# the completed household file.  These plans will be duplicated in the
# plan file from the restarted Router.  This script will remove the
# duplicate plans from the original file.

# plan1 is the plan file from the aborted router, plan2 is the plan
# file from the restarted router, and dest plan is where the combined
# plan file should go.

# Get list of travelers in each plan file
awk '{print $1}' RS='\
\
' $1 | uniq | sort -n | uniq > /tmp/trav1

awk '{print $1}' RS='\
\
' $2 | uniq | sort -n | uniq > /tmp/trav2

# Get list of duplicate travelers
cat /tmp/trav[12] | sort -n | uniq -d > /tmp/trav_dup

# Remove the duplicates
$TRANSIMS_HOME/bin/IndexPlanFile $1
$TRANSIMS_HOME/bin/RemovePlans $1 $3 /tmp/trav_dup
cat $2 >> $3

#rm -f /tmp/trav[12] /tmp/trav_dup
```

2.3 CreatePartialHHFile

```
#!/usr/local/gnu/bin/perl -w

# This script takes a household file and a completed household file,
# and creates a new household file which contains the households not
# in the completed household file. This is used to restart failed
# routers.

# Usage: CreatePartialHHFile <hhfile> <completed hh file> <new household file>

if ($#ARGV != 2)
{
    die "Usage: $0 <hhfile> <completed hh file> > <new household file>\n";
}

$hhfn = $ARGV[0];
$chhfn = $ARGV[1];
$new_hhfn = $ARGV[2];

%hhids = ();
open $hhfile, $hhfn;
while (<$hhfile>)
{
    $hhids{$_}=1;
}
close $hhfile;
my $chhfile;

unless (open $chhfile, $chhfn)
{
    print "Can't open $chhfn\n";
    exit;
}

while (<$chhfile>)
{
    delete $hhids{$_};
}
close $chhfile;

unless (open $new_hhfile, ">".$new_hhfn)
{
    print "Can't open $new_hhfn\n";
    exit;
}
```

```
foreach $hh (sort keys %hhids)
{
    print $new_hhfile "$hh";
}
```

2.4 TravToHH.pl

```
#!/usr/local/gnu/bin/perl -w

if ($#ARGV != 1)
{
    die "Usage: $0 <trav file> <hh file>\n";
}

$travfile = $ARGV[0];
$hhfile = $ARGV[1];
$travhhmapfile = "/home/transims/CaseStudy3/scenarios/allstr/population/trav_hh.txt";

# read map file
open $mf, $travhhmapfile;
while (<$mf>)
{
    ($trav, $hh) = split ' ', $_;
    $map{$trav} = $hh;
}
close $mf;

open $tf, $travfile;
while (<$tf>)
{
    ($trav) = split ' ', $_;
    if (defined($map{$trav}))
    {
        $hh=$map{$trav};
        $hhs{$hh} = 1;
    }
    else
    {
        print "Unknown traveler id $trav\n";
    }
}
close $tf;

open $hf, ">$hhfile";
foreach $hh (sort keys %hhs)
{
    print $hf "$hh\n";
}
close $hf;
```

2.5 distrib.sh

```
#!/bin/csh
set CFG=$1
set i=0
set cpu=76
set NP=47

if ( $#argv != 1 ) then
    echo "Usage: $0 <config file>"
    exit
endif

while ($i < $NP)
    echo $i rp$cpu
    set result=`ping -q -c 2 -w 5 rp$cpu | grep "0 packets received"`
    if ( "X$result" == "X" ) then
        ssh -f rp$cpu time /home/transims/CaseStudy3/bin/DistributePlans $CFG $i >& dist.$i.errs
        @ i++
    else
        echo "Skipping rp$cpu"
    endif
    @ cpu++
end
```

2.6 mkindex.sh

```
#!/bin/csh

if ( $#argv != 1 ) then
    echo "Usage: $0 <run id>"
    exit
endif

#set i=66
set i=76
set DIR=/home/transims/CaseStudy3/scenarios/allstr/plans/$1
set BIN=/home/transims/CaseStudy3/bin

foreach f ($DIR/RS*.A?)
    set result=`ping -q -c 2 -w 5 rp$i | grep "0 packets received"`
    if ( "X$result" == "X" ) then
        echo "rp$i $f"
        ssh -f rp$i time $BIN/IndexPlanFile $f >& $f.index.errs
    else
        echo "Skipping rp$i"
    endif
    @ i++
end
```

2.7 mvplans.sh

```
#!/bin/csh

if ( $#argv != 2 ) then
    echo "Usage: $0 <dir> <run id>"
    exit
endif

set DIR=/home/transims/CaseStudy3/scenarios/allstr/plans/$1
set BIN=/home/transims/CaseStudy3/bin
#set CF=/home/transims/CaseStudy3/scenarios/allstr/config_files/allstr_CA.cfg
#set CF=$3

mkdir -p /tsa/$1
cd /tsa/$1
cp $DIR/*.idx .
cp $DIR/*.t?? .

ls *.t?? > names
sed 's/$/.valid/g' < names > names.valid

foreach f (`cat names`)
    echo `pwd` >! tmp
    echo $f >> tmp
    $BIN/IndexFileNames $f.valid.trv.idx r tmp
    $BIN/IndexFileNames $f.valid.tim.idx r tmp
end

time $BIN/CatIndices plans.$2.FB `cat names.valid` >& ci1.errs &
time $BIN/CatIndices -f plans.$2.FB `cat names.valid` >& ci2.errs
```

2.8 valididx.sh

```
#!/bin/csh

if ( $#argv != 2 ) then
    echo "Usage: $0 <run id> <config file>"
    exit
endif

set DIR=/home/transims/CaseStudy3/scenarios/allstr/plans/$1
set BIN=/home/transims/CaseStudy3/bin
#set CF=/home/transims/CaseStudy3/scenarios/allstr/config_files/allstr_CA.cfg
set CF=$2
set cpu=66

foreach f ($DIR/plans.RS*.FB.t??)
#foreach f ($DIR/RS*.??)
    set i=0
    while ($i != $cpu)
        set result=`ping -q -c 2 -w 5 rp$cpu | grep "0 packets received"`
        if ( "X$result" == "X" ) then
            echo "rp$cpu $f"
            ssh -f rp$cpu time $BIN/PlanFilter -v $CF -o $f.valid $f >&! $f.valid.errs
            @ cpu++
            set i = $cpu
        else
            echo "Skipping rp$cpu"
            @ cpu++
        endif
    end
end
```